

PARALLEL ALGORITHM FOR CALCULATING GENERAL EQUILIBRIUM IN MULTIREGION ECONOMIC GROWTH MODELS

Nikolai B. Melnikov

Lomonosov Moscow State University;
Central Economics and Mathematics Institute, RAS, Moscow, Russia
melnikov@cs.msu.ru

Arseniy P. Gruzdev

Lomonosov Moscow State University, Moscow, Russia
gruzdev@cs.msu.ru

Michael G. Dalton

National Oceanic and Atmospheric Administration, Seattle WA, USA
michael.dalton@noaa.gov

Brian C. O'Neill

National Center for Atmospheric Research, Boulder CO, USA
Email: boneill@ucar.edu

Abstract: We develop and analyze a parallel algorithm for computing a solution in a multiregion dynamic general equilibrium model. The algorithm is based on an iterative method of the Gauss–Seidel type and exploits a special block structure of the model. Calculation of prices and input-output ratios in production for different time steps is carried out in parallel. We implement the parallel algorithm using the OpenMP interface for systems with shared memory. The efficiency of the algorithm is studied with the numbers of cores varying in the full range from one to the number of time steps of the model.

Key words: Computable general equilibrium, Economic growth, Iterative methods, High-performance computing, OpenMP.

AMS Classification: 91B50, 91B62, 91B66, 91B74, 68W10

1. Introduction

Dynamic computable general equilibrium (CGE) models are widely used for estimating the effects of demographic and technological changes on energy use and carbon dioxide (CO₂) emissions. The equilibrium is described in the framework of the Arrow–Debreu theory, which leads to a systems of nonlinear equations. Usually large-scale nonlinear systems are solved by one of the “general-purpose” Krylov subspace solvers, which can deal effectively with sparse matrices (see, e.g., [1]).

In our paper [2], we presented a parallel algorithm based on an iterative method of the Gauss–Seidel type [3]. We exploited the special block structure of the nonlinear system of equations in dynamic CGE models. We implemented the algorithm using parallel programming environments for the one-region version of the Population-Environmental-Technology (PET) model [4, 5]. The

numerical results showed that the speed of our algorithm is comparable to the one of Krylov methods solvers such as NITSOL [6].

In this paper we extend the algorithm to models with international trade and apply it to the multiregion PET model [7]. We implement the parallel algorithm using the OpenMP interface for systems with shared memory. To demonstrate the effectiveness of the parallel algorithm we use the PET model calibrated to reproduce major outcomes for the socioeconomic scenarios from the Shared Socioeconomic Pathways (SSP) database (see, e.g. [8]). The calibration of the PET model to the SSPs is described in the supplementary material to [8].

The paper is organized as follows. In Sect. 2 we present a description of the multiregion PET model. In particular, we explain in detail how the intermediate goods demand is calculated in the presence of the international trade. In Sect. 3 we present the numerical method for calculating the equilibrium and explain the parallel algorithm. In Sect. 4 we discuss the calculation results.

2. Structure of the CGE model

In this section we describe the multiregion PET model (for description of the one-region PET model, see, e.g. [2, 4, 5]).

The PET model is a forward-looking CGE model with three types of agents: consumers, producers, and government. Consumers maximize their lifetime utility function taking prices as given (Subsec. 2.1). Producers maximize profits supported by the prices as described in Subsec. 2.2. Government redistributes capital through taxes and transfers (for details see, e.g. [5]). International trade is described by the Armington model as described in Subsec. 2.3. Prices are determined by the markets clearing conditions for production factors, intermediate and final goods (Subsec. 2.4). The first-order optimality conditions for the agents and supply-equals-demand conditions for markets form a system of nonlinear equations. A solution to this system of equation is called the *general equilibrium*.

2.1. Consumers side

In each of the N_R regions the utility function of the representative household is given by the discounted lifetime consumption

$$U(c) = \frac{1}{\psi} \sum_{t=0}^{\infty} \beta^t n_t \left(\sum_{j=1}^{N_C} (\mu_{jt} c_{jt})^\rho \right)^{\frac{\psi}{\rho}},$$

where $t = 0, 1, 2, \dots$ is time, index $j = \overline{1, N_C}$ labels consumer goods, c_{jt} is consumption, n_t is the size of population, $\psi \in (-\infty, 1) \setminus \{0\}$ is the intertemporal substitution parameter, $\beta \in (0, 1)$ is the discount rate, $\sigma = 1/(1 - \rho)$ is the elasticity ($\rho \in (-\infty, 1) \setminus \{0\}$ is the substitution parameter) and μ_{jt} is the preference coefficient (for details of calculating μ_{jt} see, e.g., [5]).

The capital dynamics is

$$(1 + \nu_t) k_{t+1} = (1 - \delta) k_t + x_t, \quad (2.1)$$

where k_t is capital ($k_0 > 0$), x_t is investment, $\delta \in (0, 1)$ is the capital depreciation coefficient, $1 + \nu_t = n_{t+1}/n_t$ is the population growth coefficient (ν_t is the growth rate).

The budget constraint is

$$\sum_{j=1}^{N_C} p_{jt} c_{jt} + q_t x_t = (1 - \theta_t) \omega_t l_t + (1 - \phi_t) r_t k_t + g_t, \quad (2.2)$$

where p_{jt} is the price of the j th consumer good, q_t is the prices of investments, ω_t is the wage rate, r_t is the rental rate of capital, g_t is the government transfers, l_t is the labor supply, θ_t and ϕ_t are the tax rates on capital and labor incomes, respectively. Here the quantities c_{jt} , k_t , x_t , l_t and g_t are given in *per capita* terms.

Taking prices as given, the representative household maximizes the utility,

$$U(c) \rightarrow \max, \quad (2.3)$$

subject to constraints (2.1) and (2.2). The first-order optimality condition for problem (2.1), (2.2) and (2.3) gives the Euler equation

$$\frac{q_t}{\bar{p}_t} \bar{c}_t^{\psi-1} = \beta \frac{q_{t+1}(1-\delta) + (1-\phi_{t+1})r_{t+1}}{\bar{p}_{t+1}} \bar{c}_{t+1}^{\psi-1},$$

where the consumption composite and price index are

$$\bar{c}_t = \left[\sum_{j=1}^{N_C} (\mu_{jt} c_{jt})^\rho \right]^{\frac{1}{\rho}}, \quad \bar{p}_t = \left[\sum_{j=1}^{N_C} \left(\frac{p_{jt}}{\mu_{jt}} \right)^{\frac{\rho}{\rho-1}} \right]^{\frac{\rho-1}{\rho}}$$

such that

$$\sum_{j=1}^{N_C} p_{jt} c_{jt} = \bar{p}_t \bar{c}_t.$$

The transversality conditions

$$\lim_{t \rightarrow \infty} \lambda_t k_t = 0,$$

where λ_t is the Lagrange multiplier, guarantees that the optimal trajectory (c_t, k_t, x_t) exists and is unique (see, e.g., [9]).

2.2. Producers side

Firms are aggregated into sectors that produce final goods (N_C consumer goods and one “investment good”) and intermediate goods (N_E energy goods and the rest, which we call materials). The total number of production sector is $N_X = N_C + 1 + N_E + 1$.

Production level of the good X is defined by the constant elasticity of substitution (CES) function

$$X = \gamma_X (\alpha_K (G_K K)^{\rho_X} + \alpha_L (G_L L)^{\rho_X} + \alpha_{\bar{E}} (G_{\bar{E}} \bar{E})^{\rho_X} + \alpha_{\widehat{M}} (G_{\widehat{M}} \widehat{M})^{\rho_X})^{\frac{1}{\rho_X}}, \quad (2.4)$$

where K is capital, L is labor, \bar{E} is energy composite and \widehat{M} is materials (unlike small letters that indicate the *per capita* values, capital letters denote the totals). Here G_I , $I = K, L, \bar{E}, \widehat{M}$, are the productivity factors and the coefficient γ_X normalizes the production shares α_I to unity. Both productivity factors and production shares can be sector- and time-dependent. (Current version of the PET model [8] also has land as a production factor but, for simplicity, we do not consider it here.)

At each time moment, the producer of the good X maximizes profit, or equivalently, minimizes costs

$$P_K K + P_L L + P_{\bar{E}} \bar{E} + (1 + \tau_{\widehat{M}}) P_{\widehat{M}} \widehat{M} \rightarrow \min_{K, L, \bar{E}, \widehat{M}}, \quad (2.5)$$

given the level of production (2.4). Here P_I is the corresponding price and $\tau_{\widehat{M}}$ is the tax on the use of materials (for brevity, we omit the time index).

The minimal cost for problem (2.4) and (2.5) is given by $P_X X$, where

$$P_X = \frac{1}{\gamma_X} \left(\alpha_K^{\frac{1}{1-\rho_X}} \left(\frac{P_K}{G_K} \right)^{\frac{\rho_X}{\rho_X-1}} + \alpha_L^{\frac{1}{1-\rho_X}} \left(\frac{P_L}{G_L} \right)^{\frac{\rho_X}{\rho_X-1}} + \alpha_{\bar{E}}^{\frac{1}{1-\rho_X}} \left(\frac{P_{\bar{E}}}{G_{\bar{E}}} \right)^{\frac{\rho_X}{\rho_X-1}} + \alpha_{\widehat{M}}^{\frac{1}{1-\rho_X}} \left(\frac{(1+\tau_{\widehat{M}})P_{\widehat{M}}}{G_{\widehat{M}}} \right)^{\frac{\rho_X}{\rho_X-1}} \right).$$

The cost minimizing input-output ratios $A_X^I = I/X$ for $I = K, L, \bar{E}$ are given by

$$A_X^I = \left(\frac{1}{\alpha_I(\gamma_X G_I)^{\rho_X}} \frac{P_I}{P_X} \right)^{\frac{1}{\rho_X-1}},$$

and for $I = \widehat{M}$ the ratio is given by

$$A_X^{\widehat{M}} = \left(\frac{1}{\alpha_{\widehat{M}}(\gamma_X G_{\widehat{M}})^{\rho_X}} \frac{(1+\tau_{\widehat{M}})P_{\widehat{M}}}{P_X} \right)^{\frac{1}{\rho_X-1}}.$$

Since the PET model is primarily intended for energy economics analysis, it is detailed in the energy sector,

$$\bar{E} = \gamma_{\bar{E}} \left(\sum_{i=1}^{N_E} \alpha_{E_i} (G_{E_i} E_i)^{\rho_{\bar{E}}} \right)^{\frac{1}{\rho_{\bar{E}}}}, \quad (2.6)$$

where E_i , $i = \overline{1, N_E}$ are different energy types. Solving the cost-minimization problem

$$\sum_{i=1}^{N_E} (1 + \tau_{E_i}) P_{E_i} E_i \rightarrow \min_{E_i}$$

given the level of production (2.6), we derive the price of the energy composite,

$$P_{\bar{E}} = \frac{1}{\gamma_{\bar{E}}} \left(\sum_{i=1}^{N_E} \alpha_{E_i}^{\frac{1}{1-\rho_{\bar{E}}}} \left(\frac{(1+\tau_{E_i})P_{E_i}}{G_{E_i}} \right)^{\frac{\rho_{\bar{E}}}{\rho_{\bar{E}}-1}} \right)^{\frac{\rho_{\bar{E}}-1}{\rho_{\bar{E}}}}$$

and the input-output ratios $A_{\bar{E}}^{E_i} = E_i/\bar{E}$,

$$A_{\bar{E}}^{E_i} = \left(\frac{1}{\alpha_{E_i}(\gamma_{\bar{E}} G_{E_i})^{\rho_{\bar{E}}}} \frac{(1+\tau_{E_i})P_{E_i}}{P_{\bar{E}}} \right)^{\frac{1}{\rho_{\bar{E}}-1}},$$

where τ_{E_i} , $i = 1, \dots, N_E$, are the taxes on the use of energy.

2.3. Intermediate goods demand

Production has a nested structure. Therefore, calculation of the intermediate goods demand requires a recursive procedure. We derive the necessary formulae first for the one-region model and then for the multiregion case.

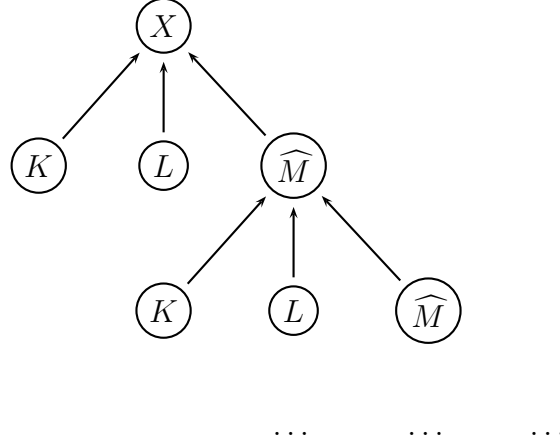


Figure 1. Nested production structure with one intermediate good.

2.3.1. One-region case

To explain the main idea, we first consider the market in which intermediate goods are aggregated into one good, which we call materials \widehat{M} . In this case, according to the nested production structure shown in Fig. 1, demand for materials is given by

$$\widehat{M} = A_X^{\widehat{M}} X + A_{\widehat{M}}^{\widehat{M}} (A_X^{\widehat{M}} X) + (A_{\widehat{M}}^{\widehat{M}})^2 (A_X^{\widehat{M}} X) + \dots,$$

where the first term corresponds to the portion of materials used in production of the final good $X(K, L, \widehat{M})$, the second term corresponds to the portions of materials used in production of materials $\widehat{M}(K, L, \widehat{M})$ one level down, etc. Calculating the sum of the geometric series, we obtain

$$\widehat{M} = \left(1 - A_{\widehat{M}}^{\widehat{M}}\right)^{-1} A_X^{\widehat{M}} X \quad (2.7)$$

or, equivalently,

$$\widehat{M} = A_X^{\widehat{M}} X + A_{\widehat{M}}^{\widehat{M}} \widehat{M}. \quad (2.8)$$

The latter means that demand for materials is equal to the amount of materials needed to produce the final good and amount needed to produce the materials themselves. Denoting $Z = \widehat{M}$, $A = A_{\widehat{M}}^{\widehat{M}}$ and $Y = A_X^{\widehat{M}} X$, we write (2.7) as

$$Z = (1 - A)^{-1} Y. \quad (2.9)$$

Next, we consider the production (2.4) with two intermediate goods, energy and materials.

In this case, the aggregate demand for materials is given by

$$\begin{aligned} \widehat{M} &= A_X^{\widehat{M}} X + \left[A_{\widehat{M}}^{\widehat{M}} A_X^{\widehat{M}} X + A_{\widehat{E}}^{\widehat{M}} A_X^{\widehat{E}} X \right] \\ &\quad + \left[A_{\widehat{M}}^{\widehat{M}} \left(A_{\widehat{M}}^{\widehat{M}} A_X^{\widehat{M}} X + A_{\widehat{E}}^{\widehat{M}} A_X^{\widehat{E}} X \right) + A_{\widehat{E}}^{\widehat{M}} \left(A_{\widehat{M}}^{\widehat{E}} A_X^{\widehat{M}} X + A_{\widehat{E}}^{\widehat{E}} A_X^{\widehat{E}} X \right) \right] + \dots \end{aligned}$$

This formula describes the sum over layer of the nested production structure (Fig. 2). Each expression in square brackets corresponds to a particular layer. Rearrangement of the terms in square brackets gives

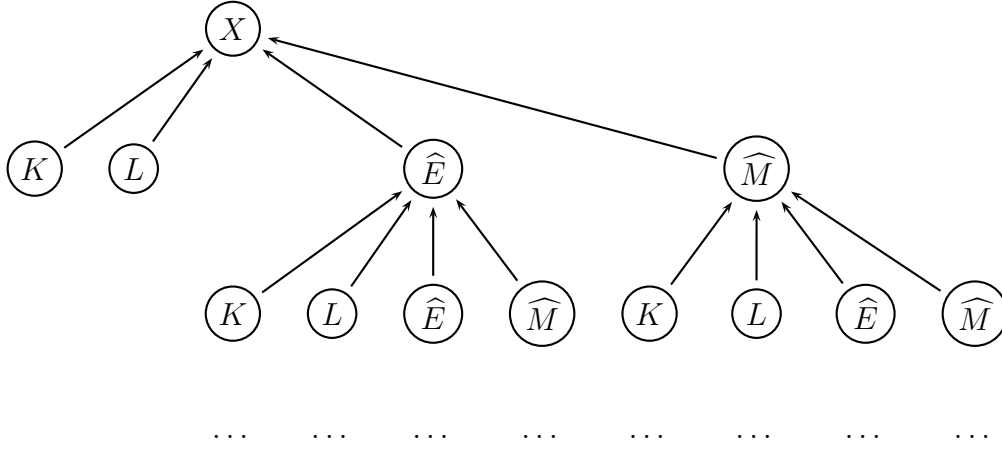


Figure 2. Nested production structure with energy and materials.

$$\begin{aligned} \widehat{M} &= A_X^{\widehat{M}} X + \left[A_M^{\widehat{M}} A_X^{\widehat{M}} X + A_E^{\widehat{M}} A_X^{\widehat{E}} X \right] \\ &+ \left[\left(A_M^{\widehat{M}} A_M^{\widehat{M}} + A_E^{\widehat{M}} A_M^{\widehat{E}} \right) A_X^{\widehat{M}} X + \left(A_M^{\widehat{M}} A_E^{\widehat{M}} + A_E^{\widehat{M}} A_E^{\widehat{E}} \right) A_X^{\widehat{E}} X \right] + \dots \end{aligned} \quad (2.10)$$

Similarly, for energy we obtain

$$\begin{aligned} \widehat{E} &= A_X^{\widehat{E}} X + \left[A_M^{\widehat{E}} A_X^{\widehat{M}} X + A_E^{\widehat{E}} A_X^{\widehat{E}} X \right] \\ &+ \left[\left(A_M^{\widehat{E}} A_M^{\widehat{M}} + A_E^{\widehat{E}} A_M^{\widehat{E}} \right) A_X^{\widehat{M}} X + \left(A_M^{\widehat{E}} A_E^{\widehat{M}} + A_E^{\widehat{E}} A_E^{\widehat{E}} \right) A_X^{\widehat{E}} X \right] + \dots \end{aligned} \quad (2.11)$$

Defining $y = (A_X^{\widehat{M}} X, A_X^{\widehat{E}} X)^T$ and

$$A = \begin{pmatrix} A_M^{\widehat{M}} & A_E^{\widehat{M}} \\ A_M^{\widehat{E}} & A_E^{\widehat{E}} \end{pmatrix},$$

we write expressions (2.10) and (2.11) as a matrix series:

$$\begin{pmatrix} \widehat{M} \\ \widehat{E} \end{pmatrix} = (I + A + A^2 + A^3 + \dots) \begin{pmatrix} A_X^{\widehat{M}} X \\ A_X^{\widehat{E}} X \end{pmatrix},$$

where I is the unity 2×2 -matrix. Summing the series, we have

$$\begin{pmatrix} \widehat{M} \\ \widehat{E} \end{pmatrix} = (I - A)^{-1} \begin{pmatrix} A_X^{\widehat{M}} X \\ A_X^{\widehat{E}} X \end{pmatrix}. \quad (2.12)$$

Equation (2.12) can be written in the form

$$\mathbf{Z} = (I - A)^{-1} \mathbf{Y}, \quad (2.13)$$

where

$$\mathbf{Z} = \begin{pmatrix} \widehat{M} \\ \widehat{E} \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} A_X^{\widehat{M}} X \\ A_X^{\widehat{E}} X \end{pmatrix}.$$

Note that equation (2.13) is the same as equation (2.9) we obtained with one intermediate good. It is the dimensionality of this equation and form of the vectors \mathbf{Z} and \mathbf{Y} and matrix A that change when we change the number of intermediate goods.

2.3.2. Multiregion case

In this subsection we obtain the intermediate goods demand in the multiregion economy with trade.

International trade is described by the Armington model (see, e.g. [10]). It is based on the assumption that the same goods produced in different regions are not perfect substitutes but can be aggregated according a certain rule (usually a CES function). The Armington model enables the representation of markets in which domestically produced goods keep a share of domestic markets even though their price is higher than the price in other regions, and in which different exporters co-exist even if they have different prices.

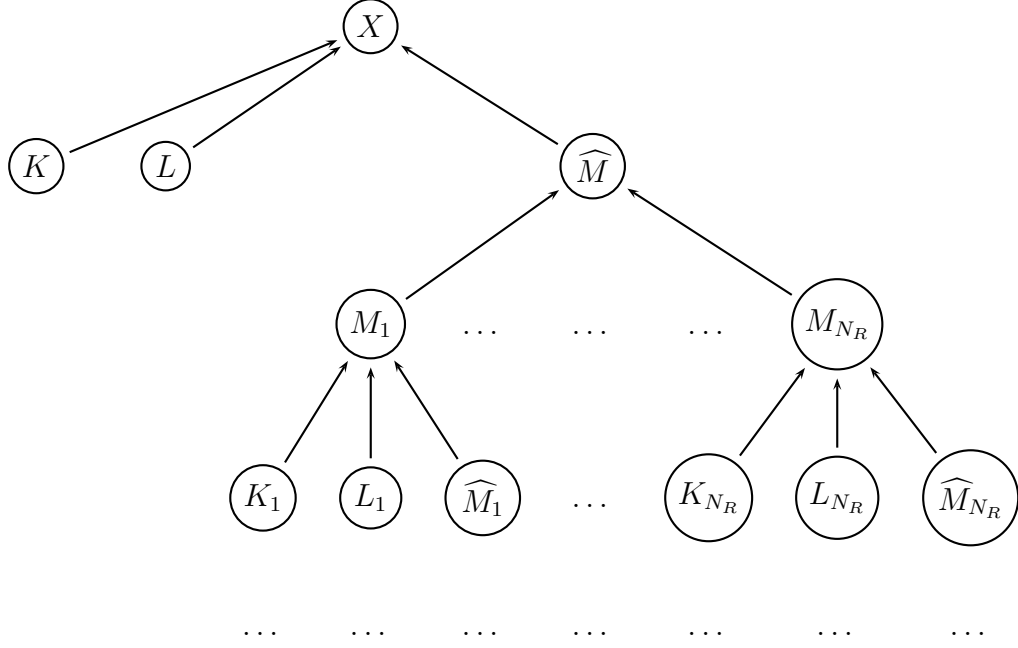


Figure 3. Nested production structure with one intermediate good for the multiregion case.

Same as in the previous subsection, first we consider the market with only one intermediate good (Fig. 3). Then $\widehat{M}(M_1, \dots, M_{N_R})$ aggregates materials M_1, \dots, M_{N_R} from N_R regions (Fig. 4).

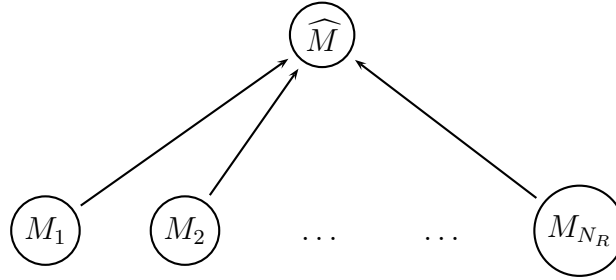


Figure 4. Armington trade structure for materials.

Similarly to the problem (2.5) and (2.4), we consider

$$P_1 M_1 + \dots + P_{N_R} M_{N_R} \rightarrow \min,$$

subject to

$$\gamma_{\widehat{M}} \left(\sum_{i=1}^{N_R} \alpha_i M_i^{\rho_{\widehat{M}}} \right)^{\frac{1}{\rho_{\widehat{M}}}} = \widehat{M},$$

where P_1, \dots, P_{N_R} are the export prices. Then the minimum of the cost function is equal to $P_{\widehat{M}} \widehat{M}$, where

$$P_{\widehat{M}} = \frac{1}{\gamma_{\widehat{M}}} \sum_{i=1}^{N_R} \left(\alpha_i^{\frac{1}{1-\rho_{\widehat{M}}}} P_i^{\frac{\rho_{\widehat{M}}}{\rho_{\widehat{M}}-1}} \right).$$

The cost minimizing input-output ratios are given by

$$b_{\widehat{M}}^{M_i} = \frac{M_i}{\widehat{M}} = \left(\frac{1}{\alpha_I \gamma^{\rho_{\widehat{M}}}} \frac{P_i}{P_{\widehat{M}}} \right)^{\frac{1}{\rho_{\widehat{M}}-1}}.$$

Similarly to relation (2.8), we have

$$\widehat{M}_i = \sum_{j=1}^{N_R} b_{ij}^M \left(A_{X_j}^{\widehat{M}_j} X_j + A_{\widehat{M}_j}^{\widehat{M}_j} \widehat{M}_j \right).$$

Denoting $B = (b_{ij}^M)$, $A^X = \text{diag} (A_{X_i}^{\widehat{M}_i})$ and $A = \text{diag} (A_{\widehat{M}_i}^{\widehat{M}_i})$, we write

$$\mathbf{Z} = (I - BA)^{-1} B \mathbf{Y} \quad (2.14)$$

where we set $\mathbf{Z} = (\widehat{M}_1, \dots, \widehat{M}_{N_R})^T$, $\mathbf{X} = (X_1, \dots, X_{N_R})^T$, $\mathbf{Y} = A^X \mathbf{X}$ and I is the unity $N_R \times N_R$ -matrix.

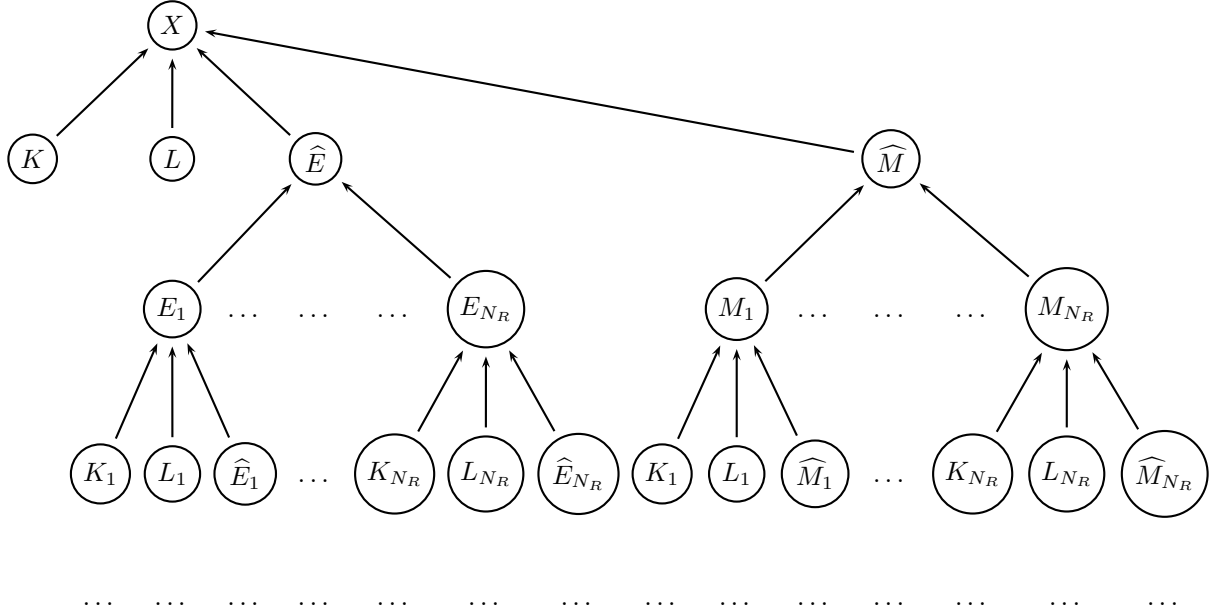


Figure 5. Nested production structure with energy and materials in the multiregion case.

In the case of production (2.4) with two intermediate goods, energy and materials (Fig. 5), the vector \mathbf{Z} has the form

$$\mathbf{Z} = (\widehat{M}_1, \widehat{E}_1, \dots, \widehat{M}_{N_R}, \widehat{E}_{N_R})^T$$

and the components of equation (2.14) will have the block structure

$$B = (B_j^i), \quad B_j^i = \begin{pmatrix} b_{ij}^M & 0 \\ 0 & b_{ij}^E \end{pmatrix},$$

where $b_{ij}^E = b_{\widehat{E}}^{E_i} / b_{\widehat{E}}^{E_j}$. Matrices A^X and A will consist of the input-output ratios for materials and energy,

$$A^X = \text{diag} (A_i^X), \quad A_i^X = \begin{pmatrix} A_{X_i}^{\widehat{M}_i} \\ A_{X_i}^{\widehat{E}_i} \end{pmatrix};$$

$$A = \text{diag} (A_i), \quad A_i = \begin{pmatrix} A_{\widehat{M}_i}^{\widehat{M}_i} & 0 \\ 0 & A_{\widehat{E}_i}^{\widehat{E}_i} \end{pmatrix}.$$

In the PET model the energy composite is the aggregate (2.6) of N_E energy types. In this case, \mathbf{Z} will be a vector of dimensions $(N_E + 1)N_R$ (N_E energy types plus materials per region). Matrix elements b_{ij}^E and $A_{\widehat{E}_i}^{\widehat{E}_i}$ will be diagonal matrices and A_i^X will have $N_E + 1$ elements.

2.4. Market equilibrium

Aggregate supply for capital K^{AS} and labor L^{AS} are determined by the sums over all regions of $n_t k_t$ and $n_t l_t$, respectively. Aggregate demand for capital and labor are

$$K^{AD} = \sum_{j=1}^{N_X} A_{X_j}^K X_j + A_{GP}^K GP,$$

$$L^{AD} = \sum_{j=1}^{N_X} A_{X_j}^L X_j + A_{GP}^L GP,$$

where GP is government purchases and A_{GP}^K and A_{GP}^L are the government sector input-output ratios of capital and labor, respectively.

An equilibrium is defined by the markets clearing conditions. That means aggregate demand is equal to aggregate supply (in each region and each time t) for the factors of production and final goods,

$$K^{AD} = K^{AS},$$

$$L^{AD} = L^{AS},$$

$$X^{AD} = X^{AS}.$$

Here X^{AD} is equal to the sums of $n_t c_t$ (or $n_t x_t$) over all regions, and X^{AS} is the production output. For the government sector, we require that revenues are equal to expenditures,

$$G^{REV} = G^{EXP}.$$

The set of the optimality conditions for consumers and producers and markets clearing conditions form a system of nonlinear equations that need to be solved. This system of equations depends on consumer quantities, i.e. capital, investment, consumption and government transfers, on the one hand and production costs (prices) and input-output ratios on the other.

3. Parallel algorithm

Since all other quantities can be obtained explicitly if we know capital K and prices P , the system of equations describing the general equilibrium can be written as

$$f(K, P) = 0.$$

The block structure of the system and parallel algorithm for solving such systems were described in detail in our paper [2]. Here we briefly recall the main ideas before describing the implementation of the parallel algorithm.

```

    input  :  $K^0, P^0$ 
    output :  $K, P$ 

1 marker:  if diff > tol and it < numIt then
2   |      omp parallel default(private)
3   |      omp shared(dyn arrays, stor arrays)
4   |      omp copyin(parameters)
5   |      omp for
6   |      for  $t \leftarrow 0$  to  $T$  do
7   |      |   Calculate prices  $P$  for time moment  $t$  (inner loop);
8   |      |   Update dyn arrays;
9   |      end
10  |      omp end parallel
11  |      Update stor arrays;
12  |      it  $\leftarrow$  (it + 1);
13  |      diff  $\leftarrow$  update ( $K, P$ );
14 end
15 goto marker

```

Figure 6. The OpenMP implementation.

The Fair–Taylor method [3] works as follows. Let K^s be the s th iterate of capital. To obtain the next iterate of prices P^{s+1} it is necessary to solve the system

$$f(K^s, P) = 0 \quad (3.1)$$

with respect to P . To obtain the next iterate of capital K^{s+2} it is necessary to solve the system

$$f(K, P^{s+1}) = 0 \quad (3.2)$$

with respect to K , and so on.

The part of the algorithm that calculates the next iterate of capital (3.2) is implemented as the *outer loop*. The part that calculates the next iterate of prices (3.1) is implemented as the *inner loop*. Blocks of the system (3.1) that correspond to different time-periods can be calculated in parallel. To improve the convergence, solution of each block is broken down into two nested loops: the NewtonA-loop for factor prices (P_K and P_L in N_R regions) and the NewtonB-loop for all other prices (goods prices in each region and export prices). The NewtonA-loop has a smaller dimensions, therefore we can use the classical Newton method with backtracking as a solver. For the NewtonB-loop we use a more advanced Krylov subspace method NITSOL (see, e.g. [6, 11]), because it has much larger dimensions and it is called more often to calculate the Jacobian for the NewtonA-loop.

The algorithm is described in Fig. 6. The input data of the algorithm is the initial approximations of capital K^0 and prices P^0 and the output is the equilibrium capital K and prices P . The general parameters are the tolerance tol and number of iterations $numIt$. Parameter T is the time horizon of the model.

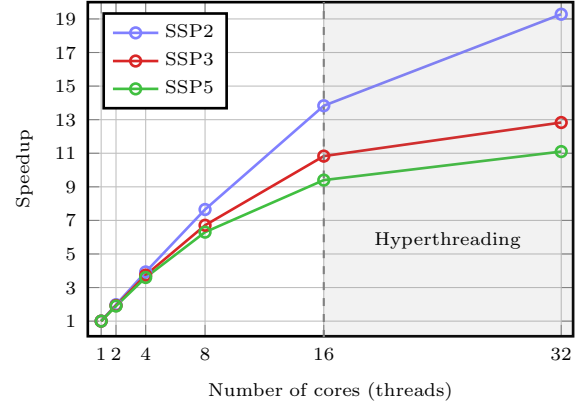
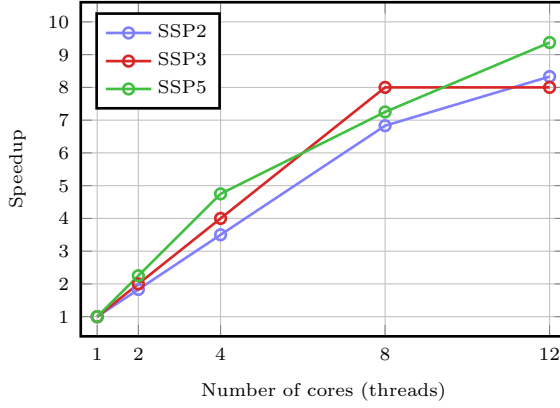
There are two types of arrays for storing and processing the economic data: *dyn* and *stor*. The first group of arrays corresponds to data at the current time and is used by the inner loop (Fig. 6, lines 6–9), the second is used for storing data over the iterations of the algorithm (outer loop). The

variable it is the iteration index and $diff$ is the target error for the outer loop. The lines 8 and 11 in Fig. 6 correspond to the implementation of economic equations and line 13 computes the error using current iterates of capital and prices.

In the OpenMP version, the time steps of the inner loop are performed in the parallel region. All dyn and $stor$ arrays are shared. The arrays with parameters are distributed using *copyin* clause (Fig. 6, line 4).

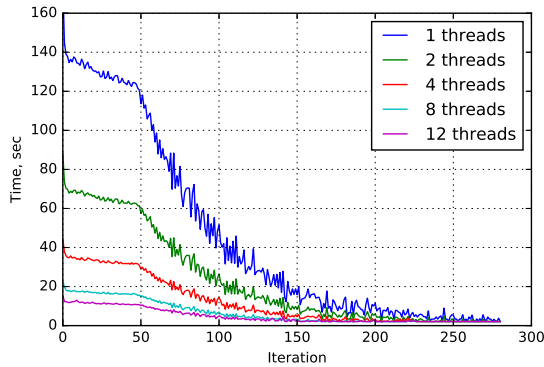
4. Results and discussion

For calculations we use the PET model with $N_R = 9$ regions and time horizon $T = 105$ years. The total number of production sectors is $N_X = 10$ in each region. As inputs the PET model uses national production and household survey data at the baseyear and long-term population and technical change projections over the whole time period. We use three sets of input data that correspond to socioeconomic scenarios from the Shared Socioeconomic Pathways (SSP) database (for the implementation of SSPs in the PET model, see [8]).

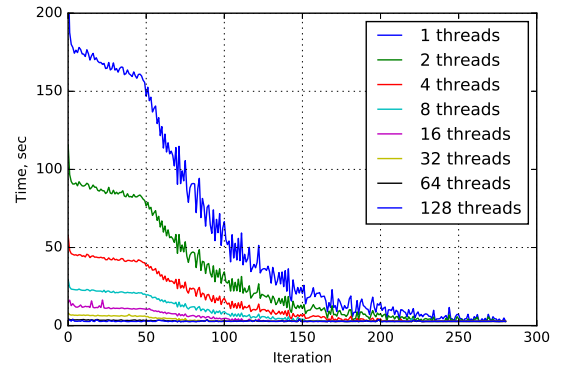


(a) Lomonosov (Intel Xeon X5670 2.93 GHz, 12 Gb) (b) Yellowstone (Intel Xeon E5-2670 2.6 GHz, 64 Gb)

Figure 7. Speedup of the model runs for different SSPs.



(a) Regular node



(b) SMP node

Figure 8. Timing of the outer loop iterations for the SSP3 obtained at the Lomonosov supercomputer.

We use two supercomputer systems for the model runs. The first one is the Lomonosov supercomputer [12]. We use two types of nodes at the Lomonosov: regular node with 12 cores (Intel

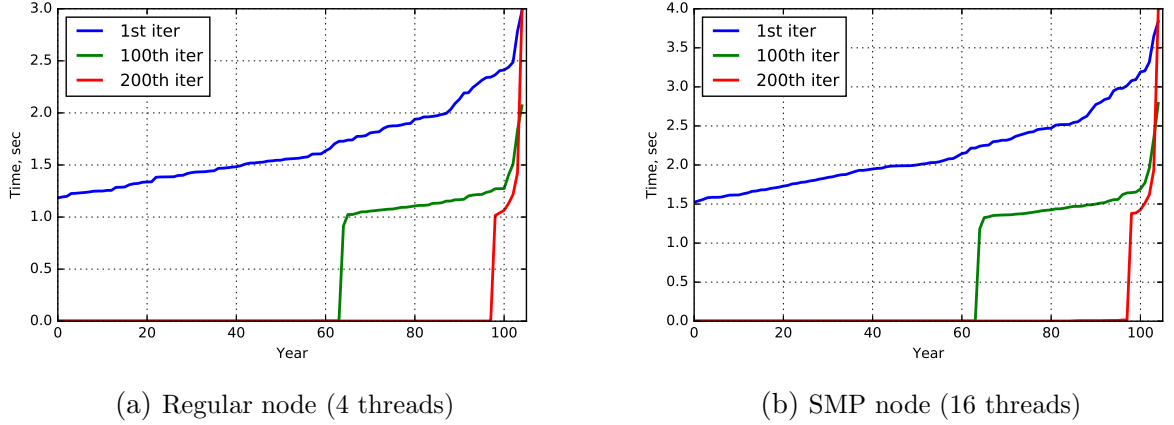


Figure 9. Timing of year-blocks in the inner loop for the SSP3 obtained at the Lomonosov supercomputer.

Xeon X5670 2.93 GHz, 1 Gb/core) and a node with 128 cores (16 Gb/core) with shared memory, the Symmetric MultiProcessing (SMP) node. The second system is the Yellowstone supercomputer [13]. At the Yellowstone we use regular node with 16 cores (Intel Xeon E5-2670 2.6 GHz, 4 Gb/core), up to 32 cores with hyperthreading. The model is implemented using Fortran. For the algorithm implementation we use BLAS [14], LAPACK [15] and Fortran implementation of NITSOL [6]. For compiling the libraries and our code we use the Intel Fortran Compiler 15 with optimization flag `-O3` and standard make-file techniques for building the project.

To study *strong scalability* of the parallel algorithm we need to increase the computing power while keeping the total problem size constant. This is achieved by running the model with the same initial approximations K_0 and P_0 and same set of numerical parameters (for each SSP) with increasing number of threads. The results show that the speedup of the parallel algorithm grows almost linearly at both supercomputers as the number of threads grows from 1 to about 12–16 (Fig. 7). Overall, we obtain the speedup of about 10 times for a regular node. With further increase of the number of nodes the speedup slows down and saturates (Table 1). Once the number of nodes becomes greater than the time horizon of the model each thread solves one year-block of the inner loop and no more speed up is possible with this algorithm. From Table 1 we see that the maximum speedup is about 22 times but using 64 nodes we already get very close to it.

Fig. 8 shows that, for the number of threads from about one to ten, there is a visible monotone decrease in timing of the outer loop as the algorithm converges (especially after the 50th iteration). This effect can be explained if we look at the timings of different year-blocks of the inner loop (Fig. 9). As the number of iterations increase the algorithm stops computing the Jacobian in the NewtonA-loop using the one from the previous iteration. The number of these “fast” year-blocks of the inner loop is increasing from iteration to iteration. For the 100th iteration of the outer loop the calculation times of more than 60 first year-blocks are close to zero. For the 200th the “fast” year-blocks span almost the whole time horizon $T = 105$.

Table 1. Speedup of the model runs for the SSP3 at the SMP node of the Lomonosov supercomputer.

Threads	1	2	4	8	16	32	64	128
Speedup	1	1.8	3.3	7.3	11	17	21.5	22

From Fig. 8 we also see that the timings of the outer loops uniformly decrease with the number of threads increasing. But as the number of threads increases above ten, the timings of the outer loops level out. The reason is that the timing of the inner loop in the parallel algorithm cannot get

smaller than the timing of the slowest year-block. From Fig. 9 we see that the number of “fast” year-blocks is increasing as the algorithm converges but there are always some “slow” year-blocks close to the end period T .

Acknowledgments

We are grateful to R. Loft and M. Weitzel for useful discussions of the results.

REFERENCES

1. **Kelley C.** Iterative Methods for Linear and Nonlinear Equations. SIAM: Philadelphia, 1995.
2. **Melnikov N., Gruzdev A., Dalton M. and O'Neill B.** Parallel algorithm for solving large-scale dynamic general equilibrium models // Russian Supercomputing Days, Moscow, 2015. P. 84–95.
3. **Fair R., Taylor J.** Solution and maximum likelihood estimation of dynamic nonlinear rational expectations models // *Econometrica*, 1983. Vol. 51. P. 1169–1185.
4. **Dalton M., O'Neill B., Prskawetz A., Jiang L. and Pitkin J.** Population aging and future carbon emissions in the United States // *Energy economics*, 2008. Vol. 30, P. 642–675.
5. **Melnikov N., O'Neill B. and Dalton M.** Accounting for the household heterogeneity in dynamic general equilibrium models // *Energy economics*, 2012. Vol. 34, P. 1475–1483.
6. **Pernice M., Walker H.** NITSOL: a Newton iterative solver for nonlinear systems // *SIAM J. Sci. Comput.*, 1998. Vol. 19, P. 302–318.
7. **O'Neill B., Dalton D., Fuchs R., Jiang L., Pachauri S. and Zigova K.** Global demographic trends and future carbon emissions // *Proc. Natl. Acad. Sci. U.S.A.*, 2010. Vol. 107, P. 17521–17526.
8. **Ren X., Weitzel M., O'Neill B.C., Lawrence P., Meiyappan P., Levis S., Balistreri E.J. and Dalton M.** Avoided economic impacts of climate change on agriculture: integrating a land surface model (CLM) with a global economic model (iPETS) // *Climatic Change*, 2016. P. 1–15. DOI: 10.1007/s10584-016-1791-1
9. **Stokey N., Lucas R. and Prescott E.** Recursive Methods in Economic Dynamics. Harvard University Press: Cambridge MA, 1989. 608 p.
10. **Armington P.** A theory of demand for products distinguished by place of production // *IMF Staff Papers*, 1969. Vol. 16, P. 170–201.
11. **Eisenstat S. and Walker H.** Globally convergent inexact Newton methods // *SIAM J. Optimization*, 1994 Vol. 4, P. 393–422.
12. **Sadovnichy V., Tikhonravov A., Voevodin V. and Opanasenko V.** “Lomonosov”: Supercomputing at Moscow State University. In *Contemporary High Performance Computing: From Petascale toward Exascale*. Chapman & Hall/CRC Computational Science, 2013. P. 283–307.
13. Computational and Information Systems Laboratory, 2012. Yellowstone: IBM iDataPlex System (Climate Simulation Laboratory). Boulder, CO: National Center for Atmospheric Research. <http://n2t.net/ark:/85065/d7wd3xhc>.
14. Basic Linear Algebra Subprograms. Available from: <http://www.netlib.org/blas/> Accessed 10 October 2016.
15. Linear Algebra Package. Available from: <http://www.netlib.org/lapack/> Accessed 10 October 2016.